

NASIS DATA BASE MANAGEMENT SYSTEM - IBM 360 TSS IMPLEMENTATION
V - RETRIEVAL COMMAND SYSTEM REFERENCE MANUAL

NASA-CR-134459) NASIS DATA BASE
MANAGEMENT SYSTEM: IBM 360 TSS
IMPLEMENTATION. VOLUME 5: RETRIEVAL
COMMAND (Neoterics, Inc., Cleveland,
Ohio.) 79 p HC \$6.00 CSCL 09B

N73-30143

Unclassified
13081

G3/08

80

NEOTERICS, INC.

prepared for



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

NASA Lewis Research Center
Contract NAS 3-14979

TABLE OF CONTENTS

I.	INTRODUCTION	4
A.	Overview.	4
B.	Ancillary Commands.	4
C.	Command Format and Notation	5
D.	Operand Representation.	5
E.	Command Description Formats	5
II.	RETRIEVAL COMMAND DESCRIPTIONS	7
A.	CANCEL Command.	7
B.	CORRECT Command	8
1.	ADD Subcommand	9
2.	CANCEL Subcommand.	9
3.	CORRECT Subcommand	9
4.	DELETE Subcommand.	10
5.	DISPLAY Subcommand	10
6.	END Subcommand	11
7.	FIELDS Subcommand.	11
8.	INSERT Subcommand.	11
9.	REPLACE Subcommand	12
10.	VERIFY Subcommand.	12
C.	DISPLAY Command	14
1.	PAGE Subcommand.	16
D.	EXECUTE Command	18
E.	EXPAND Command.	19
F.	FIELDS Command.	20
G.	FORMAT Command.	21
1.	DISPLAY Subcommand	23
2.	END Subcommand	23
3.	FIELD Subcommand	24
4.	FIELDS Subcommand.	24
5.	FORMAT Subcommand.	24
6.	FORMATS Subcommand	25
7.	HEADER Subcommand.	25
8.	NAME Subcommand.	26
9.	STORE Subcommand	27
10.	TITLE Subcommand	28
H.	FORMATS Command	30
I.	GENERATE Command.	31
J.	GPIELDS Command	32
K.	LIMIT Command	33
L.	PRINT Command	34
M.	RETRIEVE Command.	37
N.	SEARCH Command.	39
O.	SELECT Command.	41
P.	SETS Command.	52
III.	IMMEDIATE COMMAND DESCRIPTIONS	54
A.	APOFF Command	54
B.	BACK Command.	54
C.	END Command	54

D.	GO Command.	55
E.	KA Command.	55
F.	KB Command.	55
G.	PAGE Command.	55
H.	PROMPT Command.	56
I.	\$DEBUG Command.	57
J.	EXPLAIN Command.	58
K.	STRATEGY Command.	61
L.	PROFIE Command.	63
M.	SYNONYM Command.	63
N.	DEFAULT Command.	64
O.	SYNONYMS Command.	64
P.	DEFAULTS Command.	64
IV.	UTILITY COMMAND DESCRIPTIONS	66
A.	RERUN Command	66
B.	RESTART Command	66
APPENDIX A.	67
	Retrieval Command Format Summary	67
APPENDIX B.	72
	Immediate Command Format Summary	72
APPENDIX C.	74
	MT/T and Utility Command Format Summary.	74
APPENDIX D.	76
	Standard NASIS Defaults and Synonyms	76
APPENDIX E.	77
	Predefined Format Summary.	77
APPENDIX F.	78
	Glossary	78

RETRIEVAL COMMAND SUBSYSTEM REFERENCE MANUAL

I. INTRODUCTION

A. Overview

The user can operate the command subsystem conversationally or in batch mode. In conversational mode, the user interacts with NASIS while it is executing operations; he remains on-line and maintains a dialogue with the system. In non-conversational mode, the user does not maintain a dialogue with the system; the command subsystem serves as a form of job control language, since no interactive monitoring is performed by the user. The command descriptions which follow may be used in both modes of operation.

Retrieval commands can be categorized into search-oriented and output-oriented commands. The search-oriented commands employ a "browsing" capability utilizing index files for a file set (group of related files), the ability to select and combine various sets of data, and the extra capability of carrying out a user-defined linear (sequential) search on these sets of data.

The output-oriented classification of retrieval commands provides the user with the ability to review a set of data items for verification or inspection at a typewriter or CRT terminal, and to print a set of data on a remote printer. Pre-defined and user-definable data formatting is available for both output media.

B. Ancillary Commands

In addition to Retrieval commands, the Retrieval user has available to him: Immediate commands, MT/T commands, and Utility commands. Command format summaries for all of these can be found in the appendices. For greater detail, refer to the respective user's guides in Section 8, the Data Base Administrator's User's Guide.

The user can at any time, interrupt the execution of NASIS by depressing the attention/ break key at his terminal. This causes an immediate pause for the current command, and prompts the user with '-ATTN:', as opposed to '-enter:'. The user can then enter 'END' to terminate the current command, 'GO' or carriage return to continue processing

from the point of interruption, or any immediate or MT/T command to be executed.

C. Command Format and Notation

The basic format of a retrieval command is as follows:

COMMAND coperand1,operand2,...

where the COMMAND field contains a command name that identifies the requested action. The operands consist of information required by the command. The operand field may contain several operands; multiple operands must be separated by commas.

D. Operand Representation

Command operands are represented by position and keyword. The value of a specific operand may be determined from either the relative position of that operand within a series of operands or from a descriptive keyword preceding the operand value. Keyword and positional operands may appear simultaneously within the same operand field.

EXAMPLES:

```
EXPAND 30,field1
EXPAND FIELD=field1,TERM=30
EXPAND TERM=30,FIELD=field1
EXPAND 30,FIELD=field1
```

These are all equivalent expressions.

E. Command Description Formats

1. Metasymbology

Braces, (), delimit operands that may have more than one part.

```

e.g.          FORMAT          10.
(field1,field2...,fieldn)

```

Brackets, < >, delimit optional names and/or operands.

e.g. SETS <nn>

The vertical bar, |, represents an OR and

separates alternative representations of operands.

e.g. PAGE <DIRECTION=(F|B)>

Ellipses, ..., indicate that the preceding operand may be repeated one or more times.

e.g. INSERT FIELD=DATA,...

2. COMMAND Format

Upper-case letters are used in the command descriptions to indicate command names.

3. Operand Format

The word or phrase that is used to identify each operand within the operand field appears in lower-case letters. For positional operands, only the lower-case letters appear; for keyword parameters, the keyword appears in upper-case letters to the left of the equal sign and the keyword descriptor appears in lower-case.

Any character or string that is always to be written as shown appear as numbers or upper-case letters.

II. RETRIEVAL COMMAND DESCRIPTIONS

A. CANCEL COMMAND

During a retrieval session, the user may specify actions to be performed later; either later in the session, as in search, or after the session is complete, as in print. If, the user decides that these actions are not necessary, he may use the CANCEL command to eliminate them. The CANCEL command can be used to delete all pseudo-sets and PRINTs on pseudo-sets or to delete normally queued PRINTs.

Command: CANCEL
Operand: <RANGE=items>

Where:

items

identifies the items to be cancelled.

Specified as: The value 'SEARCH' to CANCEL the existing SEARCH specifications, or, a 1-3 digit numeric value(BSN), a range (two values separated by a hyphen) or '*ALL' to CANCEL all queued PRINTs.

Default: '*ALL' is assumed.

EXAMPLES:

1. The user wishes to CANCEL all queued PRINTs.

```
SYSTEM: ENTER:
USER: CANCEL
SYSTEM: ENTER RANGE:
USER: *ALL
SYSTEM: PRINT (*ALL) CANCELLED.
```

2. The user wishes to CANCEL his outstanding SEARCH specifications.

```
SYSTEM: ENTER:
USER: CANCEL
SYSTEM: ENTER RANGE:
USER: SEARCH
```

3. The user wishes to CANCEL PRINT BSN#10.

```
SYSTEM: ENTER:
USER: CANCEL 10
SYSTEM: PRINT (10) CANCELLED.
```

B. CORRECT COMMAND

The CORRECT command creates transaction records to be executed during maintenance.

Because of the multiple functions performed by CORRECT, the command itself recognizes a number of subcommands. The subcommands ADD, CANCEL, CORRECT, DELETE, DISPLAY, END, FIELDS, INSERT, REPLACE, VERIFY; are the means by which the user actually specifies the changes which he desires.

The format for the CORRECT command is:

```
Command: CORRECT
Operand: <FIELD=name> <,KEY=key> <,VERIFY=mode>
```

Where:

name: identifies the field of the record which the user wishes to examine.

Specified as: a 1-8 character data value.

key: identifies the record within the file which the user wishes to access.

Specified as: a 1-255 character data value.

mode: identifies the mode of operation for this session.

Specified as: 'YES', if the user desires an automatic display of the updated data, following each CORRECT subcommand, or 'NO', if he does not.

Sub-Commands

The subcommands give the user extensive capabilities for reviewing and correcting the data contained in a data base. The subcommands allow the user to access the records of a file, either randomly or sequentially, and to then examine the data contained in any or all of the fields of the selected record.

All data values entered as operands of CORRECT subcommands must not contain any embedded commas. Further, any leading blanks entered with an operand, are stripped off before syntax analysis. To overcome these restrictions, the user is permitted to enter operands as quoted strings. In this mode, all data within the beginning and ending quote is processed.

1. ADD Subcommand

The ADD subcommand allows the user to add a new record to the file, specifying field names and their values. The new key value is entered with its key field name. Multiple element fields can be entered as a parenthesized list. Data can be added to a null field or new elements can be added to a field. The format for the command is:

Command: ADD
 Operand: FIELD=data,<FIELD=(data,data,data),
 FIELD=data ...>

Where:

FIELD: is the 1-8 character field name.

data: is the data value to be added.

Specified as a 1-255 character data value.

2. CANCEL Subcommand

The CANCEL subcommand allows the user to nullify any corrections entered since the last CORRECT or INSERT subcommand. The format for the command is:

Command: CANCEL
 Operand: (none)

3. CORRECT Subcommand

The CORRECT subcommand allows the user to specify a new record and/or field which he wishes to examine, without returning to NASIS command mode. It provides the additional capability of accessing anchor records sequentially (forward or backward) from a given point. The format for the command is as follows:

Command: CORRECT
 Operand: <FIELD=name,><,KEY=key>

Where:

name: is the name of the next field in the record to be examined.

Specified as: a 1-8 character data value.

Default: the same field name is used.

key: is the key of the next record to be accessed.

Specified as: a 1-255 character data value, or, a signed integer value for sequential processing.

Default: the same record is used.

4. DELETE Subcommand

The DELETE subcommand allows the user to delete from the record, an element, a range of elements, a field or the entire record itself. The format for the command is as follows:

Command: DELETE
Operand: element-list

Where:

element-list: is the list of elements and/or element ranges to be deleted.

Specified as: 1) 'RECORD' to delete the entire record; 2) 'En' to delete an element (n is an integer identifying the element); 3) '(En1,En2)' to delete a range of elements (n1 and n2 are integers identifying the beginning and ending elements of the range).

Default: the entire field is deleted.

5. DISPLAY Subcommand

The DISPLAY subcommand allows the user to display the entire field. The user may 'page' sequentially through the data, or he may specify the element number at which he wishes the display to begin. The command format is as follows:

Command: DISPLAY
Operand: data

Where:

data: identifies the type of display which the user desires, sequential forward, sequential backward or positional.

Specified as: 1) 'B' for sequential backward;
2) 'En' to display from element n.

Default: display sequentially forward.

6. END Subcommand

The END subcommand allows the user to terminate processing. The format of the command is as follows:

Command: END
Operand: (none)

7. FIELDS Subcommand

The FIELDS subcommand allows the user to request a formatted display of all of the field names associated with this data base. The format of the command is as follows:

Command: FIELDS
Operand: (none)

8. INSERT Subcommand

The INSERT subcommand allows the user to specify new subfile fields for adding a new subfile record. The command format is as follows:

Command: INSERT
Operand: FIELD=data, FIELD=data, ...

Where:

FIELD: is the subfile field name to be added.

data: is the data value of the field to be added.

Specified as: a 1-255 character data value.

9. REPLACE Subcommand

The REPLACE subcommand allows the user to change data, contained in a field, by value. The format of the command is as follows:

Command: REPLACE
 Operand: start, end, old-data, new-data

Where:

start: identifies the element number at which scanning for the old data string is to begin.

Specified as: 'En' where n identifies the element desired.

Default: the current element number is used.

end: identifies the element at which scanning for the old data string is to end.

Specified as: 'En' where n identifies the element desired.

Default: the last element is used.

old-data: identifies the existing data value.

Specified as: a 1-255 character data value.

new data: identifies the replacement data value.

Specified as: a 1-255 character data value.

Default: a null value is used.

10. VERIFY Subcommand

The VERIFY subcommand allows the user to change the mode of operation. The format of the command is as follows:

Command: VERIFY
 Operand: mode

Where:

mode: identifies the subsequent mode of operation.

Specified as: 'YES' if the user desires an automatic display of the updated data, following each CORRECT subcommand, or 'NO' if he doesn't.

C. DISPLAY Command

This command displays one screen image of a data record according to a format, a set of keys (data records) according to a format, or a specific field of a data record.

The PAGE command may be used to display more screen images - either forward or backward.

Command: DISPLAY

Option 1: SET#=key
 <,FORMAT=format number|name>
 <,ITEM=items>
 <,TYPE=mode>
 Option 2: SET#=set number
 <,FORMAT=format number|name>
 <,ITEM=items>
 <,TYPE=mode>
 Option 3: SET#=set number|key
 <,FORMAT=field name>
 <,ITEM=items>
 <,TYPE=mode>

Where:

key: identifies the key of the data record to be displayed.

Specified as: a normal or quoted string that is not a set#.

Default: none; key or set# must be given.

SET#: identifies the set of keys selected by a previous SELECT command.

Specified as: nn
 nn specifies a 1 or 2 digit number between 0 and 99 inclusive.

Default: none, key or set# must be given.

FORMAT: identifies the format to govern the display output.

Specified as: nn or Fnn or fname.
 nn specifies a 1 or 2 digit sequential format number. Formats 1-5 are predefined. Formats 6 through 25 inclusive must have been specified by a FORMAT command.

Fnn specifies a 1 or 2 digit columnar format number between 1 and 25 inclusive prefixed by an F. The columnar format must have been specified by a **FORMAT** command.

fname specifies a 1 to 8 character format name as used in the **FORMAT** command **STORE** or **NAME** subcommand or the name of one of the fields of the file.

NOTE: See Appendix C for a summary of the predefined sequential formats 1 - 5.

Default: sequential format 2 for anchor key sets or sequential format 5 for subfile sets.

ITEM: specifies the first key in the set to be displayed.

Specified as: a 1-5 digit first relative key in the set.

Default: 1.

TYPE: Specifies the output mode that is to be used to display the data.

Specified as: a 1 digit code. Mode '1' means that the anchor data is to be displayed only once for each key, followed by the subfile data grouped by subfile and subfile record. Mode '2' means that the anchor data is to be repeated for each subfile record encountered, but the subfile data will still be grouped by subfile and subfile record. Mode '3' means that the anchor data is to be displayed only once for each key, but that the subfile data is to be grouped by field name and subfile.

Default: 1.

EXAMPLES: assuming previous **SELECTs** and **FORMATs**:

1. **USER:** **DISPLAY 306A68**
SYSTEM: displays the first screen of data record with key 306A68 using sequential format 2.
2. **USER:** **DISPLAY 2,F2**
SYSTEM: displays the first screen of the first record of set 2 using columnar format 2.

3. USER: DISPLAY 0,KEYFIELD,1
 SYSTEM: displays the data contained in field 'KEYFIELD' of the first record of set 0 (the file itself).

1. PAGE Subcommand

This command displays another screen image of the data being displayed.

Command: PAGE
 Operand: <DIRECTION=direction>
 <,MODE=mode>

Where:

DIRECTION: specifies the direction of paging.

Specified as: BACKWARD or FORWARD or any abbreviation thereof.

Default: forward paging.

MODE: specifies the mode:p

Specified as: S or N.

S specifies skip to the next key in the set when using a sequential format number greater than 1.

N specifies normal paging.

Default: normal paging.

EXAMPLES:

1. The user has built set 1 by a SELECT command.

USER: DISPLAY 1,3
 SYSTEM: displays the first part of the first key in set 1 using sequential format 3.
 USER: PAGE ,S
 SYSTEM: displays the first part of the second key in set 1 using sequential format 3.

2. The user wishes to examine a particular record.

USER: DISPLAY 143x,4,,3
 SYSTEM: displays the first part of the record whose key is '143x' using sequential format 4 and grouping the subfile data by

field.
USER: PAGE
SYSTEM: displays the next portion of the record
if more remains, or skips to the next
sequential record.

D. EXECUTE Command

The EXECUTE command is issued after a user has specified all of his search requests, via the SELECT command (linear search option), and is ready to execute the search which transforms pseudo-sets to sets representing actual lists of keys.

Command: EXECUTE
Operand: (none)

NOTE: During the conversational execution of a search, following an EXECUTE command, the user may depress the attention/interrupt key at his terminal to review the number of records searched and the number to be searched. He has the option to enter the END command to cancel the search. When in the normal 'ENTER' mode, the user may then enter the CANCEL command. This command has the effect of deleting all the pseudo-sets and PRINTs on pseudo-sets defined currently.

EXAMPLE:

1. The user wishes to execute a search conversationally.

SYSTEM: ENTER:
USER: EXECUTE
SYSTEM: (Upon termination of the search execution, the resolved pseudo-sets are displayed.)

E. EXPAND Command

This command displays an alphabetical sequence of cross reference (index) entries in the vicinity of a term given by the user. The PAGE command may be used to display more entries - either forward or backward alphabetically.

Command: EXPAND
Operand: TERM=term to expand,
FIELD=indexed field

Where:

TERM: specifies the value of the term.

Specified as: a normal or quoted string.

Default: none; TERM is a required argument.

FIELD

specifies the name of the indexed field.

Specified as: a 1-8 character field name.

Default: treat the anchor file as an index.

EXAMPLE:

USER: EXPAND NN,KEYWORDS
SYSTEM: Displays entries from the KEYWORDS
index near NN.
USER: PAGE
SYSTEM: Displays more entries following those
already displayed.

F. FIELDS Command

The FIELDS command is used to display the names of the data fields present on the file being used. The fields whose values have been indexed are flagged. The fields present on a subfile are grouped and displayed following a heading which identifies the subfile. If more field names exist than can be displayed on one screen, the PAGE command may be used to display more entries - either forward or backward.

Command: FIELDS
Operand: (none)

EXAMPLE:

USER: FIELDS
SYSTEM: Displays the first page of fields names.
USER: PAGE
SYSTEM: Displays the next page of field names.

G. FORMAT Command

This command establishes a current format to be further processed by subsequent FORMAT subcommands. Later on, the format may be used in DISPLAY or PRINT retrieval commands. (A list of the Predefined Formats is available in Appendix E.)

```
Command:  FCFMAT
Operand:  FNUMBER=<#><F>nn(<SP><,>NP)>>
          <,>FLDSPEC=field-specification(s)>
```

Where:

FNUMBER:
specifies the format number.

#: indicates revision of an existing format.

F: prefix distinguishes columnar format numbers from sequential format numbers.

nn: is a 1-2 digit format number from 1 to 25 or a 1-8 character name of a stored format.

SP: only applies to columnar formats.
 S indicates screen width (the default for new
 formats).
 P indicates printer width (132 columns).
 Omission indicates no change for an existing
 format.

P/NP: only applies to columnar formats.
P indicates page numbering (the default for new formats).
NP indicates no page numbering.
Omission indicates no change for an existing format.

FLDSPEC: optionally specifies new or revised field specifications.

See details under the FIELD Subcommand.

EXAMPLES:

```

USER:      format 6,(field1,field2,field3);end
SYSTEM:    builds sequential format 6.
USER:      display 3,6
SYSTEM:    displays the key field, FIELD1, FIELD2,
           and FIELD3 in sequential arrangement of

```

the first record in set 3.

```

USER:      format f7,(field1,field2(,a),
            field3(,t));end
SYSTEM:    builds columnar format F1.
USER:      print 2,f1
SYSTEM:    prints the FIELD1, FIELD2, and FIELD3 in
            columnar arrangement, with page
            numbering, a screen width wide (e.g. 40
            characters) with proportional spacing
            (e.g. 13, 12, and 12 columns plus
            truncation indicator column for the
            respective fields) of the records in set
            2 followed by the record count, the
            FIELD2 tally sum and average, and the
            FIELD3 tally.

USER:      format f2(p,np),(field1(2), field2(20,n),
            field3(30,t));end
SYSTEM:    builds columnar format F2.
USER:      print 2,F2
SYSTEM:    prints the FIELD1, FIELD2, and FIELD3 in
            columnar arrangement, without page
            numbering, 132 characters wide with
            explicit spacing (17, 9, and 103 columns
            plus truncation indicator column for the
            respective fields) of the records in set
            2 followed by the same summary
            information as in the previous example.

USER:      format #f1(,np),(field1=field4,
            field5(field2));end
SYSTEM:    revise columnar format F1: suppresses
            page numbering, replaces FIELD1 with
            FIELD4 as the field name to be output in
            the first column, inserts a new column
            for FIELD5 between FIELD4 and FIELD2, and
            re proportions the spacing (e.g. to 9
            columns plus truncation indicator columns
            each).

USER:      format #f2(s),(field2=,field5(38,t));end
SYSTEM:    revises columnar format F2: shortens the
            output width to screen width (e.g. 40
            characters) reducing FIELD3 width to 11
            plus truncation column, then deletes
            FIELD2 column increasing FIELD1 width to
            27 plus truncation column, then adds a
            column for FIELD5 with a width of 3 plus
            truncation column and element tallying
            and reduces FIELD3 to a width of 7 plus
            truncation.

```

1. DISPLAY Subcommand

This command displays the current format by simulating the effect of using the DISPLAY retrieval command with the current format.

Command: DISPLAY
Operand: (none)

Command: PAGE
Operand: <DIRECTON=b>

Where:

PAGE: command is only valid if the screen could not hold the entire display and MORE was indicated. The PAGE b command is only valid after a successful PAGE; it causes the previous screen image to be redisplayed.

EXAMPLE:

```
USER:    format #3; display
SYSTEM:  displays predefined sequential format 3
         by simulating the effect of a DISPLAY
         retrieval command.
USER:    page
SYSTEM:  displays the second screen of the
         format.
USER:    page b
SYSTEM:  re-displays the first screen of the
         format.
```

2. END Subcommand

This command is required to terminate processing of the current format.

Command: END
Operand: (none)

EXAMPLE:

```
USER:    format 7,(field1,field2,field3)
SYSTEM:  builds sequential format 7.
USER:    format 8,(field6,field5,field4)
System:  terminates processing format 7 and builds
         sequential format 8.
USER:    end
SYSTEM:  terminates processing format 8 and
         prompts for a retrieval system command.
```

3. FIELD Subcommand

This command adds, deletes, replaces, or revises the field specifications of the current format. It may not be used on predefined formats 1-5.

Command: FIELD
Operand: FLDSPEC=field-specification(s)

where if more than one field specification is entered, they must be separated by commas and the list enclosed in parentheses. Each field specification has the form:

d(<c><t>) for addition:
'e=' for deletion:
'eeee=d(<c><t>)' for replacement

Where:

d is a 1-8 character field name.
c is a 1-3 digit column position.
or an existing field name in the format.
t is a summary option (columnar only) with: T indicating tally elements, or S indicating sum elements, or A indicating tally, sum, and average elements, or any combination of three letters.
e is a 1-8 character existing field name in the format.

EXAMPLES:

See the examples under the FORMAT Command.

4. FIELDS Subcommand

This command permits the user to display the fields of his data base. It is the same as the other FIELDS command.

Command: FIELDS
Operand: (none)

5. FORMAT Subcommand

This command permits the user to start a new FORMAT without having to END his current FORMAT. It is

effectively the same as: END;FORMAT

Command: FORMAT
Operand: (same as the FORMAT Command)

6. FORMATS Subcommand

This command permits the user to display his FORMAT names without ending his current FORMAT.

Command: FORMATS
Operand: (none)

For a complete description, see the FORMATS Command described later in this text.

7. HEADER Subcommand

For a columnar format, this command specifies new or revised column heading(s) for one line at the top of each output page.

Command: HEADER
Operand: <HERLINE=line>
< ,HDRSPEC=heading specification(s)>

Where:

line specifies the relative header line.

Specified as: <#>nn<D>
prefix indicates revision on the header line,
nn indicates one or two digit header line number,
D suffix with # prefix indicates deletion of the entire header line.

Default: the next available header line

heading specification: specifies column headings for the header line. If more than one header specification is entered, they must be separated by commas and the list enclosed in parentheses.

Specified as: <text><(ename)>

where text is an alphanumeric word or a quoted string, ename is a field name already specified as a column of the output.

Default: If no heading specifications are entered with the HEADER command, each field name will be used as a column heading of each column across the header line. Otherwise, if text is omitted, the field name will be used as a column heading over the first output column for ename; if ename is omitted, text will be used as a column heading over the next successive output column to the right.

EXAMPLES: assuming a proportional format and no previous HEADER commands.

1. The user wants a default header on the first relative header line.

USER: HEADER

SYSTEM: Uses each field name as column heading of each column across the header line.

2. The user wants the text 'PARTNUM', 'ON HAND', and 'DESCRIPTION' on the third relative line of the header in the first three columns.

USER: HEADER 3,('PARTNUM', 'ON HAND', 'DESCRIPTION')

SYSTEM: generates line 2 as a blank line and put text on line 3 in appropriate columns.

3. The user now wishes to delete line 2.

USER: HEADER #2D

SYSTEM: deletes line 2 and moves line 3 up to line 2.

4. The user wants to put the text 'QUANTITY' in the next relative line of the header under the line 1 heading of 'FIELD2', and put the text 'OF PART' in the next column.

USER: HEADER ,(QUANTITY(FIELD2), 'OF PART')

SYSTEM: puts the text in the columns specified on line 3 (the next relative line.)

8. NAME Subcommand

This command assigns a name to the current format by

which the DISPLAY and PRINT retrieval commands may subsequently reference it. The name will also be used if the user subsequently STORES the format without respecifying a name.

Command: NAME
Operand: FMTNAME=format-name

Where:

format-name: is a 1-8 character alphanumeric name beginning with an alphabetic letter.

EXAMPLE:

USER: format #f1; name rpt1; end
SYSTEM: assigns RPT1 as the name of format F1.
USER: display 2,rpt1
SYSTEM: displays set 2 using format F1.

9. STORE Subcommand

This command stores the current format for future use in subsequent sessions. It may also assign a name to the current format like the FORMAT NAME subcommand does.

Command: STORE
Operand: <FMTNAME=format-name>

Where:

format-name: is only required if the current format is un-named. If specified, it is a 1-8 character alphanumeric name beginning with an alphabetic letter.

EXAMPLE:

USER: format #f2; store rpt2; end;
SYSTEM: assigns RPT2 as the name of format F2 and stores it so that it would be reinstated in a later session if the user entered, for example:

USER: print 3, rpt2
SYSTEM: prints set 3 using the previously stored format.

10. TITLE Subcommand

For a column format, this command specifies a new or revised title line for the top of each output page.

Command: TITLE
 Operand: <TTLIN=line>
 <,TTLSP=spec>

Where:

line: specifies the relative title line.

Specified as: <#>nn<D>
 # prefix indicates revision,
 nn indicates one or two digit title line
 number,
 D suffix with # prefix indicates deletion
 of the title line.

Default: The next available title line.

text: specifies the character string title value

Specified as: alphanumeric word or a quoted
 string.

Default: the current data base name.

NOTE: If necessary, blank title lines will be inserted between the last title line and a new title line.

EXAMPLES: assuming no previous TITLE commands on the current format.

1. The user wants a default title centered on the first relative title line.

USER: title

2. The user wants 'POPULAR RECORDS' as a title on the third relative title line.

USER: title 3,'POPULAR RECORDS'
 SYSTEM: inserts a blank second title line and
 centers the text on the third title
 line.

3. The user decides to eliminate the blank second title line.

USER: title #2D
 SYSTEM: deletes blank second title line and moves
 third title line up to second.

4. The user wants 'IN 1971' as a title under 'POPULAR RECORDS.'

USER: title , 'IN 1971'
 SYSTEM: centers the text as a new third title.

5. The user wants to replace the second title line with a different title.

USER: title #2, 'RECORDS MOST RETRIEVED BY
 ENGINEERS'
 SYSTEM: replaces the second title line with the
 new text centered.

H. FORMATS Command

The FORMATS command is used to display for the user a list of all output formats available. The list will be alphabetically sequenced and the predefined sequential formats will be distinguished from the stored formats by a leading asterisk. (A list of the Predefined Formats is available in Appendix E.) Named formats will be so identified. Unnamed formats will be identified as Fnn or nn depending upon whether they are columnar or sequential. If more names exist than can be shown on one screen, the PAGE command may be used to see more screens - either forward or backward.

Command: FORMATS
Operand: (none)

EXAMPLE:

USER: formats
SYSTEM: Displays the first page of format names.
USER: page
SYSTEM: Displays the next page of format names.

I. GENERATE Command

The GENERATE command is used to perform the special manipulations of key lists available to generic key data bases. The command takes into account the tree-like structure of the data base keys and gives the user the ability to modify a list of keys to correspond to a higher or lower level in the structure. This may either expand or contract the list.

Command: GENERATE

Operands: FIFID=key level field mnemonic,

SET=set number or key

Where:

key level field mnemonic: specifies the relative level within the generic key structure, at which keys are to generated.

set number or key:

specifies the key or key list which is to be used to derive the generated keys.

EXAMPLE:

```
USER:    generate home,282-34-3956
SYSTEM:  1 2 HCMF'S OF 282-34-3956
USER:    generate owner,1
SYSTEM:  2 1 COWNER'S OF 1.
```

J. GFIELDS Command

The GFIELDS command is used to display the mnemonic values assigned to the various key field levels of generic data bases. The values shown can then be used in subsequent GENERATE commands to manipulate lists of generic keys.

Command: GFIELDS
Operand: (none)

EXAMPLE:

USER: gfields
SYSTEM: Displays the mnemonic values of the generic key field levels.

K. LIMIT Command

This command is used to limit the list of keys contained in an existing set of anchor file keys according to specified criteria, thereby creating a new set.

Command: LIMIT
Operand: SET#=set-number,
 LIMTEST=(fieldname=value1:value2(,...))

Where:

SET#: denotes the number of a set consisting of anchor file keys. A list of subfile keys is unacceptable.

Default: none; SET# is a required argument.

LIMTESTS: specifies the anchor key subfield name and the value range (criteria) used to determine whether or not a key from the input set is to appear in the resultant set. Any number of tests may be specified, separating each by commas. If only one value is specified, then the test becomes an equal criteria rather than a between criteria.

DEFAULT: none; LIMTESTS is a required argument.

NOTE: If multiple tests are specified, then for an anchor key to be acceptable, it must pass each and every one of the tests.

EXAMPLE:

USER: LIMIT 1,(YEAR=65:72,CLCUDCOV=0.20)
SYSTEM: Creates a new set consisting of those key from set 1 which have an assigned date from 1965 to 1972 and whose cloud covers in less than 20%.

L. PRINT Command

This command prints, a key according to a format, or all or some of a set of keys according to a format.

1. a key according to a format.
2. all or some of a set of keys or records of those keys according to a format.

Command: PRINT

Option 1: SET#=key
 <,FCRMAT=format number>
 <,ITEMS=items>
 <,TYPE=mode>
 <,CCPIES=number of copies>
 <,SCREEN=screen mode>

Option 2: SET#=(set number|S-number)
 <,FORMAT=format number>
 <,RANGE=range>
 <,TYPE=mode>
 <,CCPIES=number of copies>
 <,SCREEN=screen mode>

Where:

key: identifies the key of the data record to be printed.

Specified as: a normal or quoted string that is not a set# or an S-number.

set: identifies the set or pseudo-set (S-number) selected by a previous SELECT command.

Specified as: nn or Sxx
 nn specifies a 1 or 2 digit set number between 0 and 99 inclusive.

Sxx specifies a 1 or 2 digit pseudo-set number between 1 and 25 inclusive prefixed by an S.

FORMAT: identifies the format to govern the printed output.

Specified as: nn or Fnn or fname.
 nn specifies a 1 or 2 digit sequential format number. Formats 1-5 are predefined. Formats 6 through 25 inclusive must have been specified by a

previous FORMAT command.

fnn specifies a 1 or 2 digit columnar format number between 1 and 25 inclusive prefixed by an F. The columnar format must have been specified by a previous FORMAT command.

fname specifies a 1 to 8 character format name as used in the FORMAT command STORE or NAME subcommand.

NOTE: See Appendix C for a summary of the predefined sequential formats 1 - 5.

Default: sequential format 2 for anchor key sets or sequential format 5 for subfile sets.

RANGE: specifies the range of keys in the set to be printed.

Specified as: <m><-n>

m is a 1-5 digit first relative key in the set to be printed. '*ALL' May be used to print all of the keys in the set.

-n is a 1-5 digit last relative key in the set to be printed prefixed by a hyphen.

Default: if RANGE is omitted, the default is all the keys in the set. If only **m** is given, **n** is defaulted equal to **m**, i.e., only the **m**th key will be printed. If only **-n** is given, **m** is defaulted to 1, i.e., the first **n** keys will be printed.

TYPE: Specifies the output mode that is to be used to print the data.

Specified as: A one digit code. Mode '1' means that the anchor data is to be displayed only once for each key, followed by the subfile data grouped by subfile and subfile record. Mode '2' means that the anchor data is to be repeated for each subfile record encountered but the subfile data will still be grouped by subfile and subfile record. Mode '3' means that the anchor data is to be displayed only once for each key, but that the subfile data is to be grouped by field name and subfile.

Default: 1

COPIES: Specifies the number of copies to print.

Specified as: A one digit code (1-9).

SCREEN: Specifies the screen mode or output size.

Specified as: A one position code. Mode 'S' means print with the default SCRNWTH. Mode 'P' means print with the maximum printer width (132).

EXAMPLES: assuming previous SELECTs, and FORMATS:

1. USER: print 306A68
SYSTEM: prints key (thus, data record)
306A68 using sequential format 2
2. USER: print 2,F2
SYSTEM: prints all the keys (thus, data
records) in set 2 using columnar
format 2.

M. RETRIEVE Command

NASIS retrieval is invoked by entering the RETRIEVE command. Once the RETRIEVE command has finished executing, the user will be prompted for the command that he wishes to execute next. At this point, the user has all of the NASIS retrieval commands available to him, including the RETRIEVE command itself, if he wishes to begin a new session, perhaps using some other file.

Command: RETRIEVE FILE, NAME, ADDRESS, SECURITY
 Operand: FILE=name of desired file,
 NAME=user's name,
 ADDRESS=user's mailing address
 SECURITY=<'Y'or'N'>

Where:

FILE: is the name of the file set from which the user wishes to retrieve information during this session. If the file name is omitted when the RETRIEVE command is entered, the default value (if one exists) will be used. If no default value exists the user will be prompted to enter the file name. If the user then defaults the file name, the RETRIEVE command will be terminated. prompted for it. If the user then defaults the file name, the RETRIEVE command will be terminated.

NAME: is the name that is used in the user's mailing address. It is a character string of from 1 to 50 characters long. If it is omitted, the default value (if one exists) will be used. If no default value exists the user will be prompted for it. If the user then defaults, no name will be generated for him.

ADDRESS: is the user's mailing address. It is a character string of from 1 to 100 characters long. The user's address combined with the user's name make up the complete user's mailing address. This is the address to which all listings generated on the high-speed printer will be mailed. If it is omitted, the default value (if one exists) will be used. If no default value exists then the user will be prompted for it. If the user then defaults, no address will be generated for him.

SECURITY: is used to indicate whether the user wishes to specify a security code for this session. This code allows the user access to secured fields with a matching code. If he replies 'y', he will be

prompted for it and his input will be masked.
This parameter is optional.

EXAMPLES:

1. The user wishes to enter the NASIS system accessing a file and use the default values for HEIGHT and WIDTH.

```
SYSTEM: ENTER NASIS COMMAND:
USER:  retrieve
SYSTEM: FILES AVAILABLE FOR RETRIEVAL:
        (shows all files the user has been
        permitted.)
SYSTEM: SELECT FILE:
USER:  (enter selected file number)
SYSTEM: ENTER YOUR NAME:
USER:  PHIL PRITCHARD
SYSTEM: ENTER YOUR ADDRESS:
USER:  CHEM LAB, LEWIS RESEARCH CENTER
SYSTEM: ENTER:
```

2. The user has been running a retrieval session and now wishes to begin a new session using a different file and specifying a security code.

```
SYSTEM: ENTER:
USER:  retrieve FILEB,,,Y
SYSTEM: ENTER SECURITY CODE
        (If possible, the system provides a
        masked area upon which the user may type
        his security code.)
SYSTEM: ENTER:
```

N. SEARCH Command

SEARCH is used to enter linear search command strings which are then passed to the SELECT command.

```
Command:  SEARCH
Operand:  EXPR= expression,
          FIELD= field name,
          REPLACE=S#,
          METHOD='SEARCH'
```

Where:

EXPR: is a boolean expression made up of references to previously defined sets and new criteria.

Specified as: (see SELECT EXPRESSION Specifications)

FIELD: is a field-name which defines the field to be used for comparison when a value appears in the expression with no associated field-name.

REPLACE: is an s# which is to be replaced with the current expression.

METHOD: is used to force a search even if the field in the expression is indexed. To do this, METHOD must be equal to 'SEARCH'.

After the SEARCH command is entered the user is prompted for search criteria:

```
Prompt:    SELECT (set# or S#) IF:
Response:  EXPR=expression
           <FIELD=field name>
           <REPLACE=S#>
           <METHOD='SEARCH'>
```

Where:

"set#" or "s#" is the result of evaluation of the expression entered with the SEARCH command.

"EXPR", "FIELD", "REPLACE", and "METHOD" as defined for the SELECT and SEARCH commands.

EXAMPLE:

The user has created set number 1 by a previous SELECT and he now wishes to perform a linear search on this set.

```
SYSTEM: ENTER:
USER: SEARCH
SYSTEM: ENTER EXPRESSION:
USER: 1
SYSTEM: SELECT 1 IF:
```

At this point, the user may enter his linear search request. (Refer to the writings on the SELECT command.) Once the user has entered his linear search option, it will be processed and then the user will be prompted for his next search request by the prompt:

```
SELECT 1 IF:
```

At this time, the user may enter his next linear search request on set 1.

When the user has entered all of his linear search requests on set 1 and wishes to terminate the search command, a default to the prompt:

```
SELECT 1 IF:
```

will return to NASIS command control which will prompt the user for his next Retrieval command.

Note: Any valid SELECT expression is valid as a response to the SELECT term IF: prompt.

The SEARCH command will take a full boolean expression just as the SELECT command does. The result of the expression (set# or S#) will be used for the term in the SELECT term IF: prompt.

EXAMPLE:

```
SYSTEM: ENTER:
USER: SEARCH '32',EMPAGE
SYSTEM: 1 12 '32',EMPAGE
        SELECT 1 IF:
```

The user now enters his linear search request as above.

0. SELECT Command

The SELECT Command takes as input an expression describing the user's criteria for selection of records from the data base. The final result is a list of keys (called a set) pointing to those data records which meet the stated criteria.

```
Command:  SELECT
Operand:  EXPR=expression,
          <FIELD=field-name,>
          <REPLACE=S#,>
          <METHOD='SEARCH'>
```

Where:

EXPR: is a boolean expression made up of references to previously defined sets and new criteria.

Specified as: The following is a detailed description, with examples, of the SELECT expression and the elements from which the expression is composed.

1. DIGIT (0,1,2,3,....,9)
2. LETTER (a,b,c,...x,y,z,\$,_,#,@)
3. ALPHANUMERIC (Any DIGIT or LETTER)
4. CHARACTER (Any character)
5. BOOLEAN-OPERATORS (in order of precedence)

Intersection (AND)	& or *
But not (NOT)	~ or -
Union (OR)	or +
6. MAPPING-OPERATORS

Children	C/control-fieldname
Parents	P/
Unique Parents	U/
Unique Set	\$/
7. RELATIONAL-OPERATORS

Greater Than	> or GT
Less Than	< or LT
Equal	= or EQ or BT or ><
Greater Than or Equal	>= or GE or -<
Less Than or Equal	<= or LE or ->
Not Equal	~= or NE or -=
Containing	<> or CN or IC
8. C1-ARRAY-OPERATORS (convention 1)

Intersection (AND)	&/ or */
Butnot (NOT)	~/ or -/
Union (OR)	/ or +/ or NULL
Individual Sets	&/

9. C2-ARRAY-OPERATORS (convention 2)

Intersection (AND)	/& or /*
Butnot (NOT)	/~ or /-
Union (OR)	/ or /+
Individual Sets	/

10. FIELDNAME must begin with a LETTER, which may be followed by 7 or less ALPHANUMERIC characters.

11. VALUE CHARACTER string in quotes, or ALPHANUMERIC string with optional quotes. Quotes will ALWAYS be necessary for, any item which resembles an E#,S#, or set# (e.g. 'En,Enn,Ennn', 'Sn,Snn','n,nn' where n represents a digit); or any item containing a delimiter (e.g. ',comma', '=equal',';semi-colon',':colon','<less than','(left paren','>greater than','right paren','|or','/slash','+plus','"quote",'&and','space','*asterisk','~not','~minus')

12. VALUE-GROUP (VALUE RELIATIONAL-OPERATOR VALUE.....)

EXAMPLE: ('31'|'32'|'33').

13. PRIMARY-TERM

a. E-number (E#) E followed by 1, to 3 digits.

EXAMPLES: E1,E101

b. Set-number (set#) 1 or 2 DIGITS. Set 0 to 99 are valid. Set 0 refers to full data base for search purposes.

EXAMPLES: 1,56

c. Pseudo-set-number (S#) S followed by 1 or 2 DIGITS. Must not be less than S1 or greater than S25.

EXAMPLES: S1,S11

d. Value VALUE. (associated
fieldname found from
second parameter
(keyword=FIELD) of
command operand, or,
FIELDNAME
RELATION-OPERATOR
(VALUE | VALUE-GROUP)

[illegible]

Note: The SELECT command will use indexed fields if possible, otherwise a linear search (to be executed by EXECUTE) is automatically invoked.

14. C1-PRIMARY-TERM-ARRAY
(used with C1-ARRAY-OPERATORS above)

E-numbers	E#:E#
Set numbers	SET#:SET#
Pseudo-set-numbers	S#:S#
Value	VALUE:VALUE OR FIELDNAME operator VALUE:VALUE where operator may be =,BT,EO,or ><

See examples under C1-ARRAY-TERM.

15. C2-PRIMARY-TEEM-ARRAY
(used with C2-ARRAY-OPERATORS above)

E-numbers	E#-E#
Set-numbers	SET#-SET#
Pseudo-set-numbers	S#-S#
Value	VALUE-VALUE or FIELDNAME operator VALUE-VALUE where operator

may be =,BT,EQ,or ><

See examples under C2-ARRAY-TERM.

16. C1-ARRAY-TERM

C1-ARRAY-OPERATOR C1-PRIMARY-TERM-ARRAY

EXAMPLES:

age='31': '42'	null operator
	implies union
+ / age='31': '42'	equivalent to above
/ E1: E10	E-number
/ 1: 13	set-number
& / S1: S5	S-number
harry: tom	implied union;
	fieldname from FIELD
author= harry: tom	implied union
/ author= harry: tom	union
@ / E1: E10	multiple sets created
@ / age='40': '50'	multiple sets created

Note: The use of the @ / operator is restricted:

Use only with E# or VALUE.

Field (for VALUE) must be indexed,

NO other terms may appear in the expression.

17. C2-ARRAY-TERM

C2-PRIMARY-TERM-ARRAY C2-ARRAY-OPERATOR

EXAMPLES:

age='31': '42' / +	operator always necessary.
E1-E10 /	E-number
1-13 / &	set-number
S1-S5 / *	S-number
harry- tom / +	fieldname from FIELD
author= harry- tom / +	equivalent to above.
E1-E10 /	multiple sets created
age='40' - '50' /	multiple sets created

Note: The use of the / operator is restricted.

Use only with E# or VALUE,

Field (for VALUE) must be indexed,

NO other terms may appear in the expression.

18. TERM

PRIMARY
or C1-ARRAY-TERM
or C2-ARRAY-TERM

19. EXPRESSION

a. boolean-expression A TERM or TERMS associated logically with BOOLEAN-OPERATORS and MAPPING-OPERATORS, with parentheses used for grouping.

EXAMPLES:

1 & 2 & 3 | 6 | '32', EMPAGE
(1 | 2 & P/KIDAGE = '3')
1 | 2 & (3 | (4 & 5))

b. S#-replacement-expression S# := BOOLEAN-EXPRESSION

EXAMPLES:

S04 :=) delete S04
S05 := CAR='FORD') Redefine S05

FIELD: is a field-name which defines the field to be used for comparison when a value appears in the expression with no associated field-name.

REPLACE: is an S# which is to be replaced with the current expression.

METHOD: is used to force a search even if the field in the expression is indexed. To do this, METHOD must be equal to "SEARCH".

The terminal output of the SELECT command is a line stating the set# or S# created, the number of keys in the set, and the input expression. If a value in the

expression is not found in the data base it is displayed within double brackets, such as:
 >>'9999'<<, AGE, and the number of references is zero.

SELECT COMMAND EXAMPLES:

```

select jones,author
select (author=jones)
select author eq jones
select 'obrian',author

select jones:morgan,author
select (author=jones:morgan)

select 1:10 & 14 |age='51')
select (age='31':'40' & ford),car
select (E1 , 1 , S1 | '1'
      | #_cars='1' | #_kids='1'),# houses

select p/1
select (p/ kidage='5')
select (1 | 2 | 6 & p/ kidage='7')
select (C/kid 1 & kidage='5')

select '31'-'40' /+,age
select '31':'40',age

```

To replace or delete an S#:

```

select #S01 delete S01
select (author='jones':'williams'),,S05
select (s05 := author='jones':'williams')
select (s23 := (null following =: causes s# to
be deleted)

```

USE OF THE SELECT COMMAND: The SELECT command, with its extension, the SEARCH command, provides the means by which the user may pick and choose, from all the information stored on the data base, those particular pieces of information in which he is interested.

The expression format of the SELECT/SEARCH command is intended to be a generalized, free-form, syntax which is easily used. It may appear difficult to learn at first due to the generality, but the user should keep in mind that most logical forms and combinations of the elements described here are valid.

1. The NASIS data base is composed of a number of records. A record is a group of data items or values which relate to a particular entity. For instance, a data base used for payroll might contain a record for each employee, and stored

within that record, would be the information describing the employee.

2. A data item or value is stored in a sub-division of the record, called a FIELD. Each field has a tag or name which the user may use to access the data within that field.
3. Each record in a data base has a unique identifier, called a KEY.
4. Just as a key uniquely identifies a single record, a group or list of keys identifies a particular group of records. Within NASIS, such a list of record keys is called a set.
5. Some fields which are referenced quite often can be indexed. That is, a list of data base record keys is stored for each value found in the field to be indexed. Logically this is equivalent to building a set for each value, however the method of storage is different.
6. Fields which are not indexed require, a linear record-by-record, search to satisfy the user's criteria and generate a set of keys.
7. As mentioned before, simple criteria or terminology which is directly convertible to a set is called term. A SELECT expression is formed when several terms are related with boolean operators (intersection, union, butnot). Using the terms listed above a sample expression would be:

```
select (1 & 2 | 3 | EMPAGE='25')
```

The SELECT command provides for a second parameter (keyword=FIELD) which contains a fieldname. This fieldname is used with all values in the expression with no fieldname directly related. Thus instead of repeating the fieldname many times the user may more easily enter:

```
select '25' | '26' | '27', EMPAGE
```

or if two fields are referenced:

```
select (car='ford'&('25'|'26'|'27')),EMPAGE
```

Value groups may also be used to avoid repeating the fieldname:

```
select (EMPAGE=('25'|'26'|'27'))
```

8. The facility to indicate an array is also provided with the use of the colon(:). The array of set numbers 2,3,4,5, and 6 may be written as:

```
2:6
```

This notation may be used with the special boolean array operators: |/, &/, and ~/.

```
|/2:6 equivalent to 2|3|4|5|6
&/2:6             2&3&4&5&6
~/2:6             2~3~4~5~6
```

If no operator is present the |/ operator is assumed. Thus

```
2:6 is equivalent to 2|3|4|5|6
```

Of course, values may also be used within this notation. Again the fieldname may appear directly associated with the values or as the second (keyword=FIELD) parameter in the SELECT command.

```
select (empage='25': '27')
select (|/empage='25': '27')
select '25': '27', empage
select |/'25': '27', empage
select '25': '26'| '27', empage
```

The five expressions above are all equivalent.

The multiple expression form of the SELECT command may be used to create several distinct sets at once:

```
select (empage='25', empage='26', empage='27')
```

yields:

SET#	#	EXPRESSION
----	---	-----
1	10	empage='25'
2	4	empage='26'
3	11	empage='27'

There is also a special array operator to convert a range of values (if the field is indexed) to a series of distinct sets:

```
select &/'25': '27' , empage
```


Which will again yield three sets as in the example in the previous paragraph.

Because the @/ operator yields multiple results, it must be the only item in an expression.

9. The SELECT command allows the use of the E# (see EXPAND Command explanation) within an expression and provides several means of converting E#'s into sets.

A single E# may comprise an entire expression:

```
select E101
```

The SELECT command will create a set and assign a unique set number. The user will receive the following message:

SET#	#REFS	EXPRESSION
----	-----	-----
1	4	EMPAGE='26'

Several E#'s may be converted to sets by using the multiple expression format:

```
select (E98,E102,E99)
```

By using a special operator (@/), a range (or array) of E#'s may be used to build sets:

```
select @/E99:E102
```

Of course boolean operators and the standard array operators may be used:

```
select E100|E101|E102
select E100:E102
```

The two expressions above are equivalent and will yield a single set containing keys for records with EMPAGE equal to 25,26, or 27.

10. If the user references a field within a subfile record, a set will be created containing subfile keys.
11. Due to differences in format, subfile key sets may not be combined directly with parent key sets in a SELECT expression. To allow conversion of subfile sets to parent sets and vice versa, operators are provided as follows:

U/ - Unique Parents of:

The subfile keys are converted to the corresponding parents keys. A parent key may appear only once in the output set. Format is: U/ term

Where term represents a subfile set.

P/ - Parent of:

The subfile keys are converted to the corresponding parent keys. One parent key will be generated for each subfile key regardless of the number of duplications. Format is: P/ term

Where term represents a subfile set.

C/control-field - Children of:

The parent keys are converted to subfile keys using the keys stored in the specified control field. Format is: C/control-field term

Where term represents a parent set, and control-field represents a control-field name, such as KID.

\$/ - Unique Subset of:

This operator removes duplications of keys from any set.
Format is:

\$/ term

Where term represents a set.

Note that \$/P/ term is equivalent to U/ term.

EXAMPLES:

```
P/KIDAGE='3'
$/1
P/'3',KIDAGE
C/KID ENIAGE='31':'40'
```

With mapping operators, subfile sets and parent sets may be combined logically using boolean operators as:

```
select (EMPAGE='31' & P/KIDAGE='36')
select (C/KID EMPAGE='31' & KIDAGE='3')
```

12. If the user enters criteria concerning a non-indexed field, then each data base record must be individually inspected. This can be very inefficient if done each time a non-indexed field is referenced, therefore the system saves the criteria for searches until the user enters the command EXECUTE, which causes all outstanding searches to be executed. This is more efficient since many tests can be made on a record once it is retrieved.

Until the search is executed, a search criteria is stored and assigned a symbolic set number or 'S#' for reference. Consider the following example where CAR is a non-indexed field:

```
select (CAR='FORD')
```

the results are:

SET#	#	EXPRESSION
----	---	-----
S01	-	(CAR='FORD')

After EXECUTE, S01 is converted to a standard set number for further reference, until that time, S01 itself can be used as a term in an expression.

Due to current restrictions only one non-indexed fieldname may appear in a single SELECT expression.

The SEARCH command is provided to give the novice user some guidance as well as allow a series of search criteria to be more easily entered.

P. SETS Command

This command displays at the user's terminal his sets or S-numbers (pseudo-sets), as specified by the parameter(s).

Command: SETS
 Operand: <nn> or
 <Snn>

Where:

nn: is the set number at which to start the display.

Specified as: nn
 nn specifies a 1 or 2 digit valid set number between 0 and 99.

Default: set number zero (0).

Snn: is the letter 'S' indicating to display S-numbers (pseudo-sets). nn is the specific S-number (pseudo-set) at which to start the display.

Specified as: a 1 or 2 digit number between 1 and 25 preceded by an S.

Default: if just the S is given, S-number 1 is assumed.

EXAMPLES:

1. If the user enters no parameters, the display will start at set number zero (0).

USER: SETS
 SYSTEM: display

2. The user may specify a specific set number.

USER: SETS 14
 SYSTEM: display

3. The user may specify S-numbers with no second parameter, defaulting to S-number 1.

USER: SETS S
 SYSTEM: display S-numbers.

4. The user may specify a specific S-number as the starting point.

USER: SETS S4
SYSTEM: display S-numbers

NOTE: When the user chooses the option of starting at a specific set number or S-number (pseudo-set) he cannot page backwards from this point. To achieve this effect, he need only restart his display at a lower number.

III. IMMEDIATE COMMAND DESCRIPTIONS

The NASIS terminal support function includes a set of immediate commands to facilitate the use of the system. The facilities provided perform functions that are executable at any time, and so, should be available from a common source.

The functions provided include task control (BACK, GO, END and APOFF), data control (KA, KB) and debugging (\$DEBUG).

A. APOFF Command

The APOFF command instructs the system to continually raise the END condition until NASIS operation is terminated. The command is invoked by entering:

APOFF

B. BACK Command

The BACK command is used to instruct the system to change the status of the task from conversational to background mode. (BACK is only valid when operating in the single terminal mode of operation). The command is invoked by entering:

BACK dsname

Where:

"dsname"

identifies the data set from which the system is to retrieve the subsequent commands.

Specified as: a 1-35 character fully qualified data set name.

Default: a data set, NASIS.BACK.SYSIN, will be created containing the commands:

/GO
/APOFF
LOGOFF

C. END Command

The END command instructs the system to terminate

the current command. The command is invoked by entering:

END

D. GO Command

The GO command is used to instruct the system to continue executing. It is normally used to resume processing following an attention interrupt. The command is invoked by entering:

GO

E. KA Command

The KA command instructs the system to begin using the full EBCDIC character set when reading from and writing to the terminal. The command is invoked by entering:

KA

F. KB Command

The KB command instructs the system to begin using the folded EBCDIC character set (automatically translate lower case alphabetic to upper case). The command is invoked by entering:

KB

G. PAGE Command

The PAGE command is used to display additional "pages" of data when the user's request generates more than one screen of information. The particular options available and the results realized are dependent on the command which generated the initial data and the nature of the data itself.

Command: PAGE

Operands: <DIRECTION=paging direction>
<,MODE=paging mode>

Where:

"paging direction"

identifies the direction of the paging operation.

Specified as: a subset or superset of the

keywords 'BACKWARD' or 'FORWARD'. If an invalid value is entered the default is used.

System Default: FORWARD is assumed.

"paging mode"
identifies the type of paging action requested.

Specified as: a subset or superset of one of the following reserved words:

"NORMAL" - page using the next available sequential data values.

"SKIP" - page from the beginning of the next item in the list (e.g. the next key in a list).

EXAMPLE:

```
USER:    DISPLAY 5,3
SYSTEM:  Displays the first screen of data
for the first key in set 5, using format 3.
USER:    PAGE
SYSTEM:  Displays the next screen of data for
the first key in the set.
USER:    PAGE ,SKIP
SYSTEM:  Displays the first screen of data
for the next key in the set.
USER:    PAGE BACK,S
SYSTEM:  Display the first screen of data for
the previous key in the set.
```

H. PROMPT Command

The PROMPT command is used to display the text and inserts (optionally) of any of the system messages whose key is known to the user. It is most commonly used to verify the existence and/or proper structure and composition of infrequently used messages.

```
Command:  PROMPT
Operands: MESSAGE=message key
          <,INSERTS=(data value,...)>
```

Where:

"message key"
identifies the message which is to be displayed to the user.

Specified as: a 1-8 character data value.

"data value"
 identifies an optional value to be inserted
 into the text of the message where
 indicated.

Specified as: 1-20 separate 1-50 character
 data values.

EXAMPLE:

```

USER:  FROMPT DBINIT04
SYSTEM:  DBINIT04  CANCELLED:  COMMAND
INVALID.
USER:  FROMPT DBINIT04,XPND
SYSTEM:  DBINIT04 CANCELLED:  COMMAND  XPND
INVALID.
  
```

I. \$DEBUG Command

The \$DEBUG command instructs the system to allow or disallow the user to enter TSS debugging statements. (\$DEBUG is valid only when operating in the single terminal mode of operation). The command is invoked by entering:

\$DEBUG mode

Where:

"mode"
 identifies the mode in which the user wishes
 to operate.

Specified as: The keyword upon "ON" if TSS
 commands are to be allowed, or "OFF" if
 not.

Default: "OFF" is assumed.

J. EXPLAIN Command

The user of the NASIS system may at times be confronted by an error message or a prompting message, whose meaning he does not fully comprehend. To assist him at times like this, NASIS has an EXPLAIN command. EXPLAIN may be entered at any time that the user is being prompted for input. The purpose for this command is to provide the user with a means for obtaining more information about a message sent to the user's terminal by the NASIS system. This information varies, depending upon the particular command format chosen, but may be, a more detailed message, the explanation of the response(s) expected (if applicable), the definition of a particular term or an indication of the program which issued the message.

The EXPLAIN command has four basic formats, which are used to indicate what type of additional information the user is requesting. To request a more detailed message, the user should enter:

EXPLAIN number

Where:

"number"

indicates the relative message number of the message for which the user is requesting an explanation.

Specified as: a signed decimal number between zero and MINUS seven.

Default: zero is assumed (current message).

A second use of the EXPLAIN command would be to obtain the explanation for a term. To request this information, the user should enter:

EXPLAIN term

Where:

"term"

identifies the term to be defined.

Specified as: a 1-8 character word which may be optionally qualified by a 1-6 character file name. The word identifies a term used by the NASIS system.

To request an explanation of the responses expected by a particular prompting message, the user should enter:

EXPLAIN RESPONSE, number

Where:

"RESPONSE"

is a keyword that identifies this form of the command.

Specified as: the 8 character term RESPONSE.

"number"

indicates the relative message number of the message for which the user is requesting an explanation.

Specified as: a signed decimal number between zero and MINUS seven.

Default: zero is assumed (current message).

The final format of the command is used to request the origin of the message. This information may be requested by entering:

EXPLAIN ORIGIN, number

Where:

"ORIGIN"

is a keyword that identifies this form of the command.

Specified as: the 6 character term ORIGIN.

"number"

indicates the relative message number for which ORIGIN the user is requesting an explanation.

Specified as: a signed decimal number between zero and MINUS seven.

Default: zero is assumed (current message).

EXAMPLES:

1. SYSTEM: -ENTER FILE NAME:
 USER: /EXPLAIN
 SYSTEM: -ENTER THE 1-6 CHARACTER NAME WHICH
 IDENTIFIES THE DATAPLEX WHOSE DESCRIPTORS
 YOU WISH TO PROCESS.
 USER: EXPLAIN DATAPLEX
 SYSTEM: DATAPLEX: THE TERM USED TO IDENTIFY THE
 COLLECTION OF PHYSICAL DATA FILES WHICH
 CONTAIN THE DATA FOR A SINGLE LOGICAL
 SET OF DATA.
 SYSTEM: -ENTER FILE NAME:
 USER: EXPLAIN ORIGIN
 SYSTEM: DEEDIT01
 SYSTEM: -ENTER FILE NAME:
2. SYSTEM: WARNING: INPUT HAS BEEN TRUNCATED
 USER: (DEPRESSES THE ATTENTION KEY.)
 SYSTEM: -ATTN:
 USER: EXPLAIN - 1
 SYSTEM: WARNING: THE DATA WHICH HAS BEEN ENTERED
 EXCEEDS THE MAXIMUM ALLOWABLE LENGTH.
 IT HAS BEEN TRUNCATED TO THAT LENGTH AND
 PROCESSING CONTINUED.
 SYSTEM: -ATTN:

K. STRATEGY Command

The strategy command is an immediate command of the NASIS system, which gives the user the ability to access the stored command sequences referred to as strategies. The strategies are built dynamically, as commands are executed, but may be listed or deleted explicitly using the strategy command. In addition, the names of all of the existing strategies may be displayed.

The strategy commands themselves are not stored in the strategy data set. If the user interrupts the strategy command by depressing the attention key, the command will be terminated.

The strategy command is invoked by entering:

STRATEGY option, strategy

Where:

"option"

Identifies the function to be performed by this invocation of the strategy command.

Specified as: one of the following keyword identifiers, LIST, DELETE, NAMES.

Default: NAMES

"strategy"

identifies the strategy to be operated upon.

Specified as: a 1-16 character name.

Default: 'CURRENT-STRATEGY'

EXAMPLES:

1. The user wishes to list the names of all strategies currently stored in his strategy data set.

SYSTEM:-ENTER:

USER : STRATEGY NAMES

SYSTEM: STRATEGIES SAVED BY JOHN PAGE 1

CURRENT-STRATEGY SAMPLE.STRATEGY

TEST WORKSTRAT1

SYSTEM:-ENTER:

2. The user wishes to list the contents of a

strategy.

SYSTEM:-ENTER:

USER : STRATEGY LIST,TEST

SYSTEM: STRATEGY TEST PAGE 1

DEFAULT SCRNMG=12,SCRNWT=50

RETRIEVE SALISR.LISR,JOHN,BUSY

REVIEW RDELCD,1,00000-99999

REVIEW RDBCORR,1,00000-99999

REVIEW REEDIT,1,00000-99999

END

SYSTEM:-ENTER:

3. The user wishes to delete a strategy.

SYSTEM:-ENTER:

USER : STRATEGY DELETE,TEST

SYSTEM:-ENTER:

L. PROFILE Command

The profile capability of the NASIS system gives the user a powerful vehicle for stream-lining and simplifying the use and operation of the system. Further, it can be used to tailor the appearance of the system, to an extent, to the particular requirements of an individual application or user.

The profile, i.e., the list of synonym and default values, is always present when the system is running, but may come from a number of sources. The usual source of the profile will be the profile data set of the TSS USERID being used. There will be a unique data set of this type for each NASISID that has requested the system to save his profile while operating under this USERID. If this has never been done, the data set will not exist, and the system will attempt to retrieve a profile from member NASISPRO of data set DBALIB(0). This member will exist if the Data Base Administrator (DBA) has established a common profile for his users. If this is not the case, the profile will be retrieved from member NASISPRO of data set LISRLIB(0), which will contain the standard system profile.

The PROFILE command is used to store the current user's profile in the profile data set for use during subsequent terminal sessions. The format of the commands is:

PROFILE

M. SYNONYM Command

The SYNONYM command is used to define new synonym values, or to modify existing synonym values for the various NASIS commands. The format of the command is:

SYNONYM value=command(...)

Where:

"value"

is the synonym value being created or modified.

Specified as: a 1-8 character alphanumeric string.

"command"

is the NASIS command for which a synonym is being defined.

Specified as: a 1-8 character alphanumeric

string.

Default: delete any entry currently existing for the value specified.

N. DEFAULT Command

The DEFAULT command is used to define new default values, or to modify existing default values for the symbolic parameters referenced by the NASIS commands. The format of the command is:

DEFAULT keyword=value(...)

Where:

"keyword"

is the symbolic name of the parameter whose default is being created or modified.

Specified as: a 1-8 character alphanumeric string.

"value"

is the default value being defined.

Specified as: a 1-255 character alphanumeric string.

Default: delete any entry currently existing for the keyword specified.

O. SYNONYMS Command

The SYNONYMS command is used to obtain a list of the current synonyms and their corresponding NASIS commands. The format for the command is:

SYNONYMS value(...)

Where:

"value"

identifies the synonym whose entry is to be displayed.

Specified as: a 1-8 character alphanumeric string.

Default: display all of the synonym entries in the profile.

P. DEFAULTS Command

The DEFAULTS command is used to obtain a list of the current defaults and their corresponding keywords. The format for the command is:

DEFAULTS keyword(,...)

Where:

"keyword"

identifies the symbolic parameter whose default is to be displayed.

Specified as: a 1-8 character alphanumeric string.

Default: display all of the default entries in the profile.

EXAMPLE:

The following example illustrates the use of the commands described above.

```
-ENTER NASIS COMMAND:
  synonym syns=synonyms,defs=defaults
-ENTER NASIS COMMAND:
  default scrnhgt=12,scrnwth=40
-ENTER NASIS COMMAND:
  syns synonyms,defaults,syns,defs;defs scrnhgt,scrnwth
    SYNONYMS FOR JCHN                      PAGE 1
SYNONYMS=SYNONYMS,8 DEFAULTS=DEFAULTS,8
SYNS        =SYNONYMS,4 DEFS        =DEFAULTS,4
    DEFAULTS FOR JCHN                      PAGE 1
SCRNHGT =12
SCRNWTH =40
-ENTER NASIS COMMAND:
  profile
-ENTER NASIS COMMAND:
```

Note: the SYNONYMS command displays the minimum number of characters that constitute a valid abbreviation for the synonym.

IV. UTILITY COMMAND DESCRIPTIONS

NASIS provides the commands RESTART and RERUN to recover from a system crash or execute previously entered commands.

- A. RERUN Command The RERUN facility is used in conjunction with the STRATEGY facility to provide the ability to sequentially execute NASIS commands entered during a previous terminal session. RERUN can be used to execute a strategy constructed by the retrieval or descriptor editor sub-system.

The RERUN command is invoked by responding to the message "ENTER NASIS COMMAND:" with:

RERUN strategy

where:

"strategy" identifies the strategy to be executed, specified as the 1-8 character name of a stored strategy.

- B. RESTART Command

The RESTART command is used in conjunction with the STRATEGY facility to provide the ability to recover from a system crash during a retrieval or descriptor editor session. Like RERUN, RESTART causes NASIS=COMMANDS SAVED IN A STORED STRATEGY TO BE EXECUTED sequentially just as if they were being entered at a terminal.

The RESTART command is invoked by responding to the message "ENTER NASIS COMMAND:" with:

RESTART

It is important to remember that following a system crash, RESTART must be the very first NASIS COMMAND ENTERED. Otherwise, the system strategy data set (CURRENT-STRATEGY), which contains all the user commands entered during the aborted session, will be scratched and a new data set created.

APPENDIX A.

RETRIEVAL COMMAND FORMAT SUMMARY

This summary maintains the following general form for each command:

- i. COMMAND
 - a. Description
 - b. Operand (defaults)
1. CANCEL
 - a. Cancel linear search.
 - b. (none)
2. CORRECT
 - a. Correct data bases online.
 - b. FIELD=name,
KEY=key,
VERIFY=mode
3. CORRECT ADD Subcommand
 - a. Add a new recrd.
 - b. FIELD=data,
FIELD=(data,data,data),
FIELD=data,...
4. CORRECT CANCEL Subcommand
 - a. Nullify corrections.
 - b. (none)
5. CORRECT CORRECT Subcommand
 - a. Specify a new record.
 - b. new-field,
new-key
6. CORRECT DELETE Subcommand
 - a. Delete data.
 - b. element-list
7. CORRECT DISPLAY Subcommand
 - a. Display entire field.
 - b. data
8. CORRECT END Subcommand
 - a. Terminate CORRECT processing.
 - b. (none)
9. CORRECT FIELDS Subcommand
 - a. Display names of fields in
current data base file.

- b. (none)
- 10. CORRECT INSERT Subcommand
 - a. Add subfile record.
 - b. FIELD=data,
FIELD=data,...
- 11. CORRECT REPLACE Subcommand
 - a. Change data by value.
 - b. start,
end,
old-data,
new-data.
- 12. CORRECT VERIFY Subcommand
 - a. Automatic display of updated data.
 - b. mode
- 13. DISPLAY (form 1)
 - a. Display set at terminal.
 - b. SET#=set number
<,FORMAT=format number{field name}>
<,ITEM#=item>
<,TYPE=mode>
- 14. DISPLAY (form 2)
 - a. Display item at terminal.
 - b. SET#=key value
<,FORMAT=format number>
<,ITEM#=items>
<,TYPE=mode>
- 15. DISPLAY PAGE subcommand
 - a. Screen manipulation.
 - b. <DIRECTON=F|B
<,MODE=S|N>>
- 16. EXECUTE
 - a. Execute linear search.
 - b. (none)
- 17. EXPAND
 - a. Expand on term.
 - b. TERM=term,
FIELD=indexed field (cross reference) name
- 18. FIELDS
 - a. Field names display.
 - b. (none)
- 19. FORMAT
 - a. Define or revise a format.
 - b. FNUMBER=<#><F>nn<(<S|P><,P|NP>)>

<,FLDSPEC=field specification(s) see FORMAT FIELD
below>

20. FORMAT DISPLAY subcommand
 - a. Display the current format.
 - b. (none)
21. FORMAT END subcommand
 - a. Terminates processing of current format.
 - b. (none)
22. FORMAT FIELD subcommand
 - a. Add, delete, replace and/or revise field specification(s).
 - b. FLDSPEC=one or more field specifications. Each field specification has the form:

dddd<(<ccc|eeee><,T|S|TS|A>)> for addition or revision, or
 eeee= for deletion, or
 eeee=dddd<(<ccc><,T|S|TS|A>)> for replacement.
23. FORMAT FIELDS subcommand
 - a. Display names of fields in the current data base file.
 - b. (none)
24. FORMAT FORMAT subcommand
 - a. Restart the FORMAT command.
 - b. FNUMBER=<#><F>nn<(<S|F><,P|NP>)>
 <,FLDSPEC=field specification(s) see FORMAT FIELD above>
25. FORMAT FORMATS subcommand
 - a. Display the names and/or numbers of currently available formats.
 - b. (none)
26. FORMAT HEADER subcommand
 - a. Add, delete or revise a header line specification.
 - b. <HDRLINE=<#>nn<D>>
 <,HDRSPEC=cne or more header specification(s)>
 Each header specification has the form:

<header text><(eeee)>
27. FORMAT NAME subcommand
 - a. Assign or replace the name of the current format.
 - b. FMTNAME=format name
28. FORMAT STORE subcommand
 - a. Store the current format for future use.
 - b. <FMTNAME=format name>

29. FORMAT TITLE subcommand
 - a. Add, delete or replace a title line specification.
 - b. <TTLLINE=<#>nn<DD>>
<,TTLSPEC=title text>
30. FORMATS
 - a. Format name display.
 - b. (none)
31. GENERATE
 - a. Modify sets based on generic key structure.
 - b. FIELD=name of a generic key level.
SET#=set number of key to be processed.
32. GFIELDS
 - a. Generic key names display.
 - b. (none)
33. LIMIT
 - a. Limit a set based on specified Criteria.
 - b. SET#=set-number,
LIMTESTS=(fieldname=value1:value2<,...>)
34. PRINT (form 1)
 - a. Print set.
 - b. SET#=set number|S-number
<,FORMAT=format number>
<,ITEMS=items>
<,TYPE=mode>
<,COPIES=number of copies>
<,SCREEN=screen mode>
35. PRINT (form 2)
 - a. Print item.
 - b. SET#=key
<,FORMAT=format number>
<,RANGE=range>
<,TYPE=mode>
<,COPIES=number of copies>
<,SCREEN=screen mode>
36. RETRIEVE
 - a. RETRIEVAL invccation.
 - b. FILE=data base name,
NAME=user's name,
ADDRESS=user's mailing address,
SECURITY=<Y|N>
37. SEARCH
 - a. Linear search on set.
 - b. SET#=(set number|S-number)
38. SELECT

- a. Select an expression.
- b. EXPR=expression
<,FIELD=fieldname>

39. SETS

- a. Sets or pseudo-set (S-number) display.
- b. <set number> cr
<S-number>

APPENDIX B.

IMMEDIATE COMMAND FORMAT SUMMARY

This appendix has been included for user convenience. For greater detail about the Immediate Command see Section 8, DBA User's Guide, Topic H.1-5.

It should be mentioned here that Immediate Commands are unique in three respects. First, they may be entered any time the user is being prompted. Second, if an Immediate Command is entered during a data prompt, it must be preceded by a slash; i.e., /END. Third, an attention/interrupt during one of these commands will result in its termination.

This summary maintains the following general form for each command:

- i. COMMAND
 - a. Description.
 - b. Operands (defaults)
-
1. \$DEBUG
 - a. Allows TSS debugging statements.
 - b. mode
 2. APOFF
 - a. Application lcgoff.
 - b. (none)
 3. BACK
 - a. Change task status to background.
 - b. dsname
 4. DEFAULT
 - a. Define or modify default values.
 - b. KEYWORD=value(,....)
 5. DEFAULTS
 - a. List current defaults.
 - b. <keyword,.....>
 6. END
 - a. Terminate the current command.
 - b. (none)
 7. EXPLAIN (form 1)
 - a. Explain relative message number.
 - b. number

8. EXPLAIN (form 2)
 - a. Explain a term.
 - b. term
9. EXPLAIN RESPONSE
 - a. Explain the expected responses for a particular prompting message.
 - b. (,number)
10. EXPLAIN ORIGIN
 - a. Explain the origin of a message.
 - b. (,number)
11. GO
 - a. Continue after an attention/interrupt.
 - b. (none)
12. KA
 - a. List the full EBCDIC character set.
 - b. (none)
13. KB
 - a. List the folded EBCDIC character set.
 - b. (none)
14. PAGE
 - a. Display additional pages of data.
 - b. <DIRECTION=(F|B)>
<,<MODE=(S|N)>
15. PROFILE
 - a. Store current user's profile.
 - b. (none)
16. PROMPT
 - a. Display test and unserts of any system message.
 - b. MESSAGE=message key
<,<INSERTS=data value(,...)>
17. SYNONYM
 - a. Define or modify a synonym value.
 - b. VALUE=command(,...)
18. SYNONYMS
 - a. List the current synonyms.
 - b. <VALUE(,...)>
19. STRATEGY
 - a. List or delete strategies.
 - b. Option,strategy-name

APPENDIX C.

MT/T AND UTILITY COMMAND FORMAT SUMMARY

This appendix was included to complete the format summary of all the commands available to the Retrieval System user.

The MT/T commands are available to only those users running under MT/T. The Utility commands are available to all users.

This summary maintains the following general form for each command:

- i. COMMAND
 - a. Description
 - b. Operands (defaults)

MT/T COMMANDS (see Section 8, Topic A.1)

1. DATETIME
 - a. Displays current date and time.
 - b. no operands
2. HELP
 - a. Request assistance from the MT/T operator.
 - b. <text>
3. MSG
 - a. Transmit a message to any other current NASIS-ID.
 - b. NASIS-ID, text
4. NUSERS
 - a. Presents number of users.
 - b. no operands
5. USERS
 - a. Presents NASISIDS currently running under NASIS.
 - b. no operands

UTILITY COMMANDS (see Section 8, Topic C.1)

1. RERUN
 - a. Rerun a stored strategy.
 - b. strategy-name
2. RESTART
 - a. Restart an abnormally terminated retrieval

- b. strategy.
- b. (none)

- 3. SECURE
 - a. Change security codes.
 - b. (none)

APPENDIX D.

STANDARD NASIS DEFAULTS AND SYNONYMS

This appendix lists the standard system defaults and synonyms assigned to each user upon JOINing NASIS. These values may be altered by the user to suit his specific needs.

DEFAULTS

ADDRESS : NASIS - DEVELOPMENT GROUP : C/O NASA
 LEWIS RESEARCH CENTER : CLEVELAND,
 OHIO
 DBAID : SADEA
 FIELD : KEYWORDS
 FILE : ASRDI
 FORMAT : 2
 ITEM : 1
 MSGIDS : N
 MSGLEVEL : W
 NAME : A NASIS USER
 OWNERID : SAOWNER
 SCRNHGT : 12
 SCRNWTH : 72
 SYSTEM : N A S I S - NASA AEROSPACE
 SAFETY INFORMATION SYSTEM
 TEXT : *** HELP ***
 TIME : 60
 USERID : OPERATOR

SYNONYMS

DEFS : DEFAULTS
 ITC : MESSAGE
 MSG : MESSAGE
 SYNS : SYNONYMS

For a listing of the all acceptable abbreviations for any command use the SYNONYMS command and specify the command name, or *ALL for all the commands.

APPENDIX E.

PREDEFINED FORMAT SUMMARY

There are five predefined sequential formats available to the NASIS user. These formats are numbered 1 - 5. Sequential formats 6 - 26 and all columnar formats must be defined by the user with the FORMAT command.

A basic description of the sequential formats 1 - 5 follows:

1. displays the key field (anchor or subfile) only. That is, the set of keys itself.
2. displays the anchor file key and all anchor file fields.
3. displays the anchor file key and all anchor file and associated file fields.
4. displays the anchor file key and all anchor file, associated file and all subfile fields.
5. display the subfile fields only.

For further information regarding the definition and use of formats see the command descriptions for: FORMAT, DISPLAY and PRINT.

APPENDIX F.

GLOSSARY

array (for the SELECT command) is a range of E-numbers, S-numbers, set-numbers, or values (for a particular field), specified by giving the beginning and ending term, such as E1:E50 or 'DICK': 'HARRY'.

CRT is a Cathode Ray Tube (television screen). In this particular document it refers to a remote computer terminal.

data base executive is the centralized data base access and security subsystem (DBPAC) for mainline programs. It is NASIS's data base input/output package.

default is a prestored value for a command parameter to be used when no explicit value is entered. The user may default when prompted by depressing the return key on his terminal.

director is a controller for a NASIS subsystem.

e-number (E#) is an element of the following form: the letter E followed by a non-negative interger. E-numbers are assigned to terms appearing in the output of the EXPAND command. The E-number can be used in the expression of a SELECT command, in place of the term.

field is a record data item having a field name, an internal field descriptor, and one or more values per record.

index file is a VISAM file which cross-references the element values of a particular anchor file field.

key is a unique character string within a file record upon which the file is sequentially organized (based upon the collating sequence of the string). Each record in the database has a unique key. When records are SELECTed by the user according to some criteria, the keys of the chosen records are saved in a list called a set.

NASIS is the NASA Aerospace Safety Information System.

pseudo-set is a list of keys represented by a pseudo-set number or S-number. A pseudo-set is unlike a set in that the list of keys it represents is not formed until some later logical point in the user's strategy at which the EXSEARCH command is issued. A pseudo-set then is transformed into a set representing an actual list of keys.

pseudo-set number (S-number) is a character string of the following form: the letter 'S' followed by an integer between 1 through 25. An S-number represents a linear search specification created through the use of the SEARCH and SELECT (linear search form) commands.

record (i.e., data record) is an anchor data set record with all of its associated data set records.

S-number (S#) is a pseudo-set number.

set is the list of anchor record keys (typically, accession numbers) associated with a term or expression.

set number (SET#) is a unique integer (1,2,3,...,97) assigned to a set during a strategy; set number 0 represents the entire anchor file for linear searches.

strategy is a sequence of NASIS commands used to perform a particular function, usually for data retrieval.

subcommand is a specialized command in support of the utilization of a command.

term (for the SELECT command) is any item which ultimately results in the formation of a set of anchor keys. It may be an E-number, S-number, set-number, or a value specified for a field, or an array of any of these. For the EXPAND command, only the value type of term is allowed.